On the Equivalence of 2-Threshold Secret Sharing Schemes and Prefix Codes

Paolo D'Arco, Roberto De Prisco, and Alfredo De Santis

Cyberspace Safety and Security, CSS 2018, Lecture Notes in Computer Science vol. 11161.

Presented by: Shannon Veitch
June 10, 2019
CS 858 Spring 2019

2-Threshold Secret Sharing Schemes

- We define a **2-Threshold Secret Sharing Scheme** (for a 1-bit secret)
- Let $\mathcal{P}$ be a set of participants, $s \in \{0, 1\}$ a secret
- A secret s is split into $n = |\mathcal{P}|$ **shares**, denoted $sh_1, \ldots, sh_n$
  - We consider a $(2, n)$-threshold scheme for finite $\mathcal{P}$
  - We consider an **evolving** threshold scheme, denoted $(2, \infty)$-threshold scheme, for infinite $\mathcal{P}$
- The following two properties should hold:
  - **Privacy:** no sh_i reveals any information about s
  - **Correctness:** a reconstruction function can be used to reconstruct s from any two sh_i, sh_j

Prefix Codes and Previous Results

- A **prefix** (or **prefix-free**) **code** is a code in which no codeword is a prefix of any other codeword.

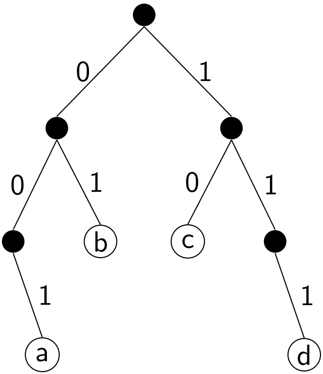
  $[10, 111, 011]$ is a prefix code

  $[1, 111, 011]$ is not a prefix code

- Prefix codes are (typically) variable-length codes

Prefix Codes and Previous Results

- A prefix code can be represented by a binary tree in which each leaf represents a codeword



character	encoding
a	001
b	01
c	10
d	111

- A **prefix code for the integers** is an infinite prefix code $C = c^1, c^2, \ldots,$ where codeword c^i encodes integer $i, i \in \mathbb{N}$

Prefix Codes and Previous Results

Theorem 1 [4]

Let $\sigma : \mathbb{N} \to \mathbb{N}$. A prefix code for the integers $C = c^1, c^2, \ldots$ such that $|c^i| = \sigma(i)$ exists if and only if it is possible to construct an evolving 2-threshold scheme for a 1-bit secret in which the size of the share for the participant is $|sh_i| = \sigma(i)$.

Proof of $\implies$:

- When participant t arrives, if necessary, extend random bitstring r to be at least $r_1 r_2 \ldots r_{|c^t|}$ bits.
- The share sh_t of participant t is defined as

$$sh_t = \begin{cases} r_1, r_2, \ldots, r_{|c^t|} & \text{if } s = 0 \\ c^t_1 \oplus r_1, c^t_2 \oplus r_2, \ldots, c^t_{|c^t|} \oplus r_{|c^t|} & \text{if } s = 1 \end{cases}$$

- Any one participant has a random bitstring, any two have two bitstrings such that either:
  - one is a prefix of the other if $s = 0$, or
  - one is not a prefix of the other if $s = 1$.

Prefix Codes and Previous Results

Theorem 1, Proof of $\Longleftarrow$:

- This direction is based on the following result:
  Theorem 2 [1]. *Let $\ell_i = |sh_i|$ be the length of the shares of a $(2, n)$-threshold secret sharing scheme, where sh_i is the share of participant i, $i = 1, 2, \ldots, n$. Then we have that $\sum_{i=1}^{n} \frac{1}{2^{\ell_i}} \leq 1$.*
- This implies that Kraft's inequality holds [2], which is a necessary and sufficient condition for the existence of a prefix code with length ℓ_i for codeword i.

Constructing Schemes from Binary Trees

- Let T be a binary tree, a a leaf on T. A **tree extension** creates two new leaves u, v, as left and right children of a, respectively.
- We label u with a random bit r and v with $s \oplus r$ where s is the secret.
- We write $(u, v) = extension(a)$.

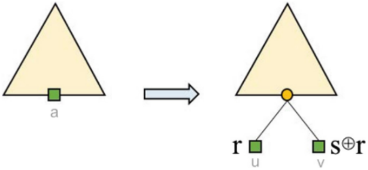


Figure: Tree extension operation

Constructing $(2, n)$-Threshold Schemes from Binary Trees

- We can associate leaves of the binary tree to participants.
- The $(u, v) = extension(a)$ operation distributes the secret to all participants rooted in a.
- Each participant will receive the label given to either u or v.

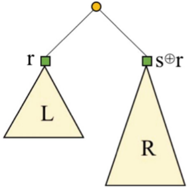


Figure: Secret split between left and right participants

- Any two participants, one belonging to L and another belonging to R can reconstruct the secret s.

Construction $(2, n)$-Threshold Schemes from Binary Trees

Theorem 2

The shares corresponding to the leaves of a binary tree with at least n leaves are a $(2, n)$-threshold secret sharing scheme.

Proof

- **Privacy:** A single participant receives a sequence of bits $b_1, \ldots, b_\ell$ where

$$b_i = \begin{cases} r_{j_i} \\ s \oplus r_{j_i} \end{cases},$$

  and each r_{j_i} is indipendent for $i = 1, \ldots, \ell$.

- **Correctness:** Two participants have shares of the form $b^1_1, \ldots, b^1_{\ell_1}$ and $b^2_1, \ldots, b^2_{\ell_2}$. Then there exists some level ℓ_0 such that $b^1_{\ell_0} = r^k_{j_{\ell_0}}$ and $b^2_{\ell_0} = s \oplus r^k_{j_{\ell_0}}$. The xor of these bits reveals the secret.

Example of a $(2, n)$-Threshold Scheme

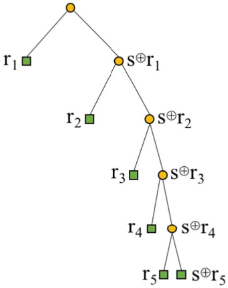


Figure: A chain-tree

Participant	Share
p_1	r_1
p_2	$s \oplus r_1, r_2$
p_3	$s \oplus r_1, s \oplus r_2, r_3$
p_4	$s \oplus r_1, s \oplus r_2, s \oplus r_3, r_4$
p_5	$s \oplus r_1, s \oplus r_2, s \oplus r_3, s \oplus r_4, r_5$
p_6	$s \oplus r_1, s \oplus r_2, s \oplus r_3, s \oplus r_4, s \oplus r_5$

- Any single p_i has no information about the secret because each random bit is independent.
- Two participants can recover s by xor-ing the appropriate bits.

Constructing $(2, \infty)$-Threshold Schemes from Binary Trees

- We can extend the previous approach to the infinite one by preserving at least one share.
- Upon arrival of a new participant, select a leaf u, not yet assigned to some p_i, and perform $extension(u)$. Assign one of the new leaves to the participant.

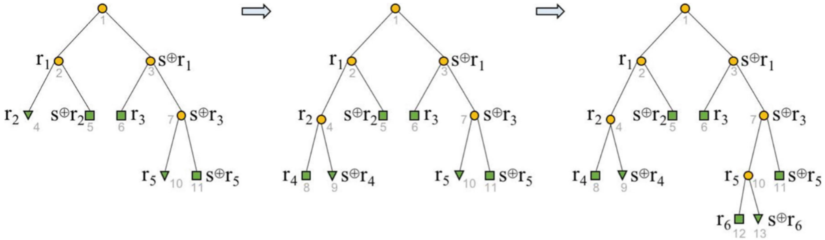


Figure: Extensions in a tree. Squares denote leaves assigned to participants. Triangle denote unassigned leaves.

Constructing $(2, \infty)$-Threshold Schemes from Binary Trees

Theorem 3

The shares corresponding to the leaves of a binary tree is a $(2, \infty)$-threshold secret sharing scheme.

Proof

The proof is the same as in Theorem 2.

Saving Randomness

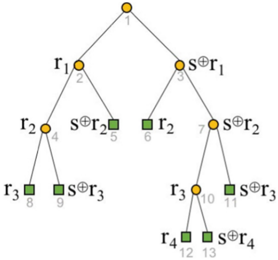


Figure: One random bit per level

- We can save randomness by using only one random bit for each level of the tree.
- Use random bit r_1 for the first level, r_2 for the second level, and so on.

Saving Randomness

Theorem 4

The shares corresponding to the leaves of a binary tree using only one random bit per level is a 2-threshold secret sharing scheme.

Proof

- **Privacy:** This is as in Theorem 2 (and 3), since each participant gets one bit per each level.
- **Correctness:** Two participants have shares of the form $b^1_1, \ldots, b^1_{\ell_1}$ and $b^2_1, \ldots, b^2_{\ell_2}$ where

$$b^k_i = \begin{cases} r_{\ell(i)} \\ s \oplus r_{\ell(i)} \end{cases} .$$

There is some level z such that for $z < s < min\{\ell_1, \ell_2\}$, $b^1_s = r_s$ and $b^2_s = s \oplus r_s$. The xor of these bits reveals the secret.

Conclusions

- A binary tree corresponds to a prefix-code and viceversa. So we have proposed an alternative approach to show the equivalence of prefix-codes and 2-threshold secret sharing schemes.
- In our construction, the size of the shares is equal to the depth of the leaves, or equivalently, to the length of the codewords.

References

[1] I.P. Cascudo, R. Cramer, and C. Xing. Bounds on the threshold gap in secret sharing and its applications. *IEEE Trans. Inf. Theory 59(9), pp. 5600-5612.* (2013).

[2] T.M. Cover and J.A. Thomas. Elements of Information Theory, 2nd edn. Wiley, Hoboken (2006).

[3] P. D'Arco, R. De Prisco, and A. De Santis. On the Equivalence of 2-Threshold Secret Sharing Schemes and Prefix Codes. *Cyberspace Safety and Security, CSS 2018, LNCS vol. 11161.*

[4] I. Komargodski, M. Naor, and E. Yogev. How to share a secret, infinitely. *TCC 2016. LNCS, vol. 9986, pp.485-514.* (2016).

Questions?